The ATTiny45

8 pins
(SOIC)

256 byte EEPROM

4KB of Flash

256 byte static RAM

Two 8-bit timer/counters with PWM and prescaler

10-bit analog-digital converter

Universal Serial Interface

.

debugWIRE

20 MIPS
throughput at 20 MHz

8MHz internal resonator

2.7V - 5.5V operating voltage
(1MHz at 1.8V)

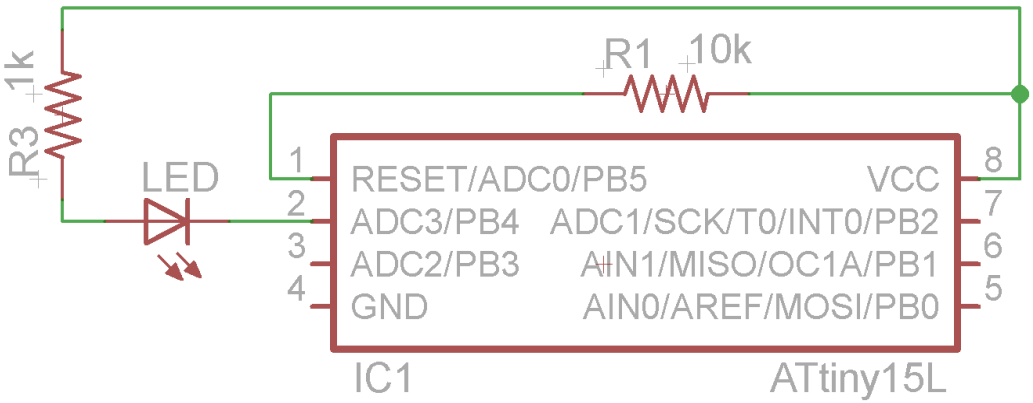
Applications

That's great and all, but what can we do with it?

Blink some LEDs

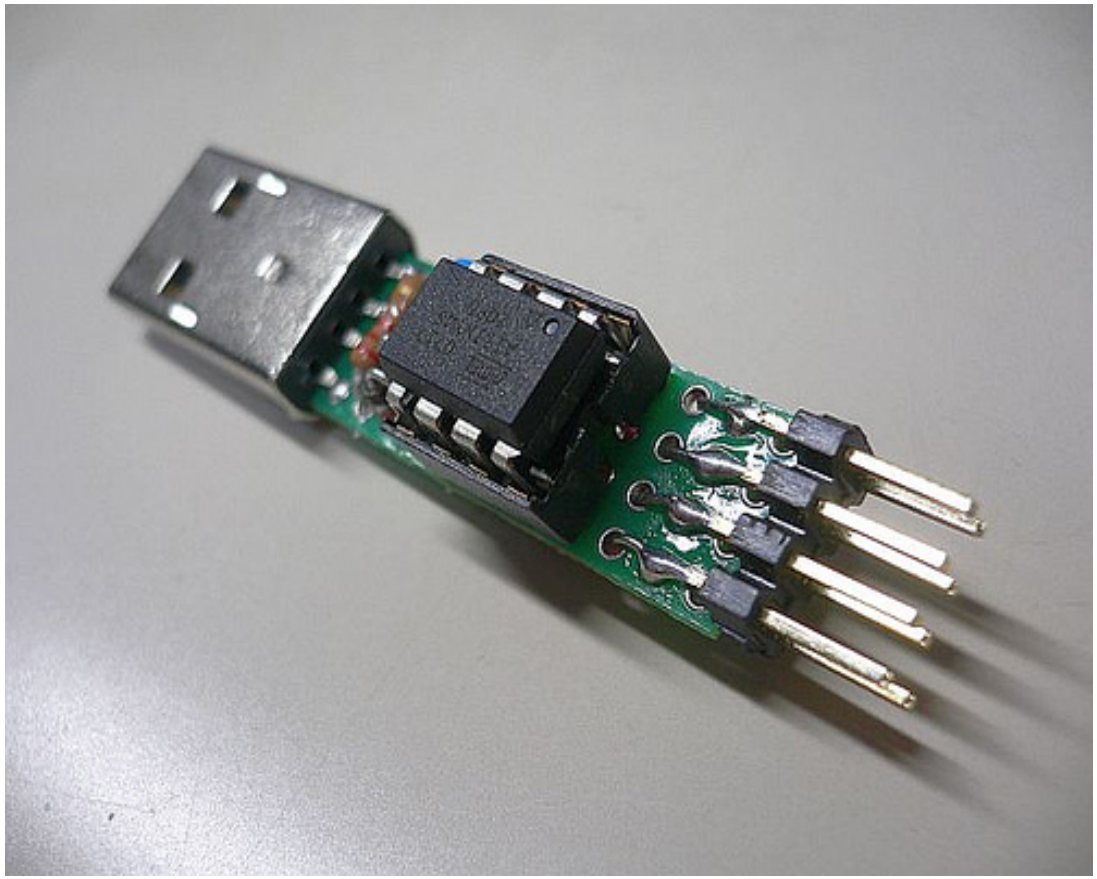
The One True Purpose of all Microcontrollers, right? Simple to set up, wire an LED to pin 3 and run:

```
;
; hello.LED.45.asm
;
; Blink an LED
;

.include "tn44def.inc"
.equ led = PB4; pin 4
.cseg
.org 0
.def temp = R17; temporary storage
.def temp1 = R18; moar temporary storage
rjmp reset

;
; main program
;

reset:
        sbi DDRB, led
        loop:
                ;start on loop
                sbi PORTB, led
                        delay1:
                                ldi temp, 255
                                delayloop:
                                        ldi temp1, 255
                                        delayloopnest:
                                                dec temp1
                                                brne delayloopnest
                                        dec temp
                                        brne delayloop
                ;start off loop
                cbi PORTB, led
                        delay2:
                                ldi temp, 255
                                delayloop0:
                                        ldi temp1, 255
                                        delayloopnest0:
                                                dec temp1
                                                brne delayloopnest0
                                        dec temp
                                        brne delayloop0
```



Onboard USB

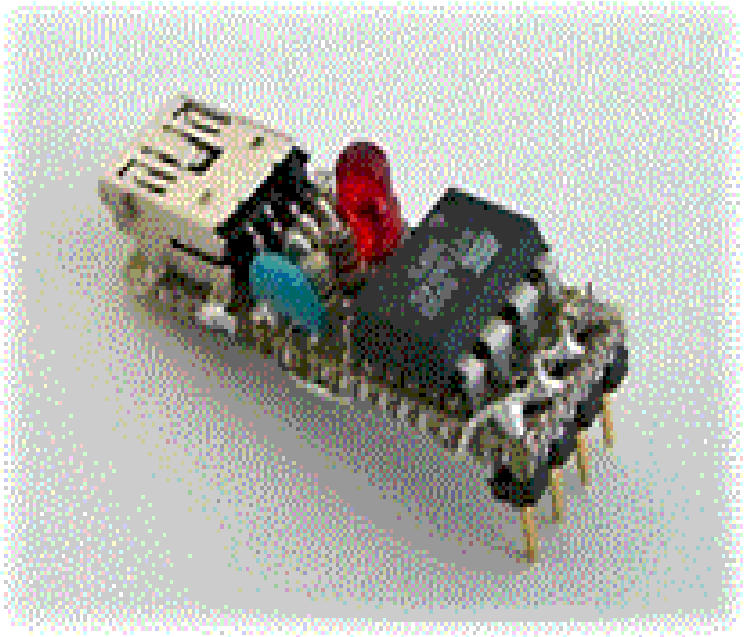
V-USB firmware lets the '45 emulate USB without a USB-serial converter

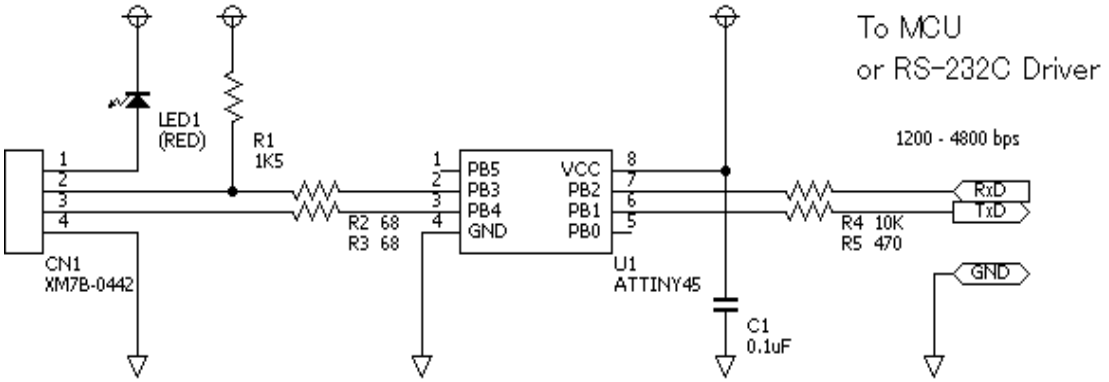


(image from tokoyama on flickr), used without permission

USB-serial converter

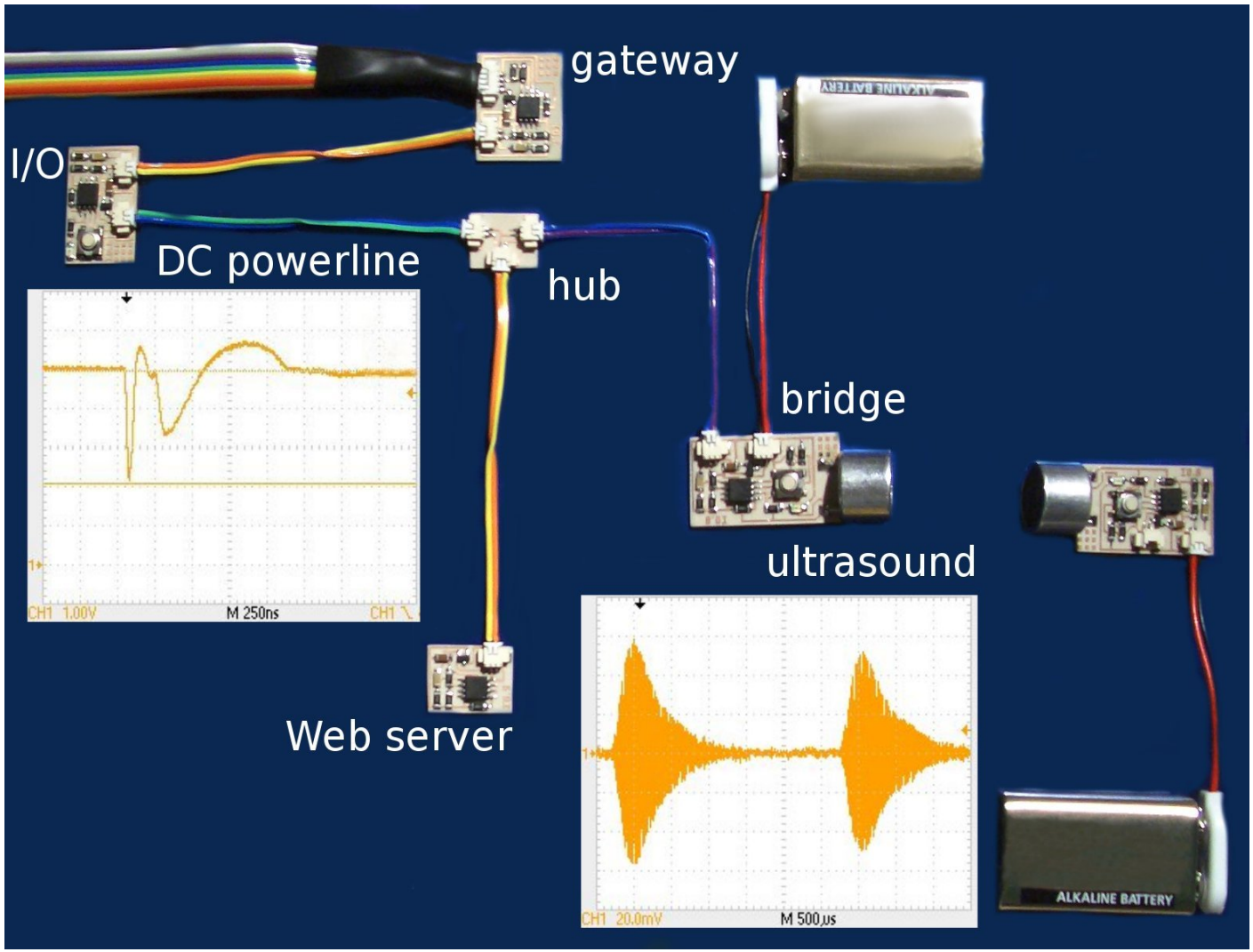
If you don't want to use a whole
2kb of flash for V-USB



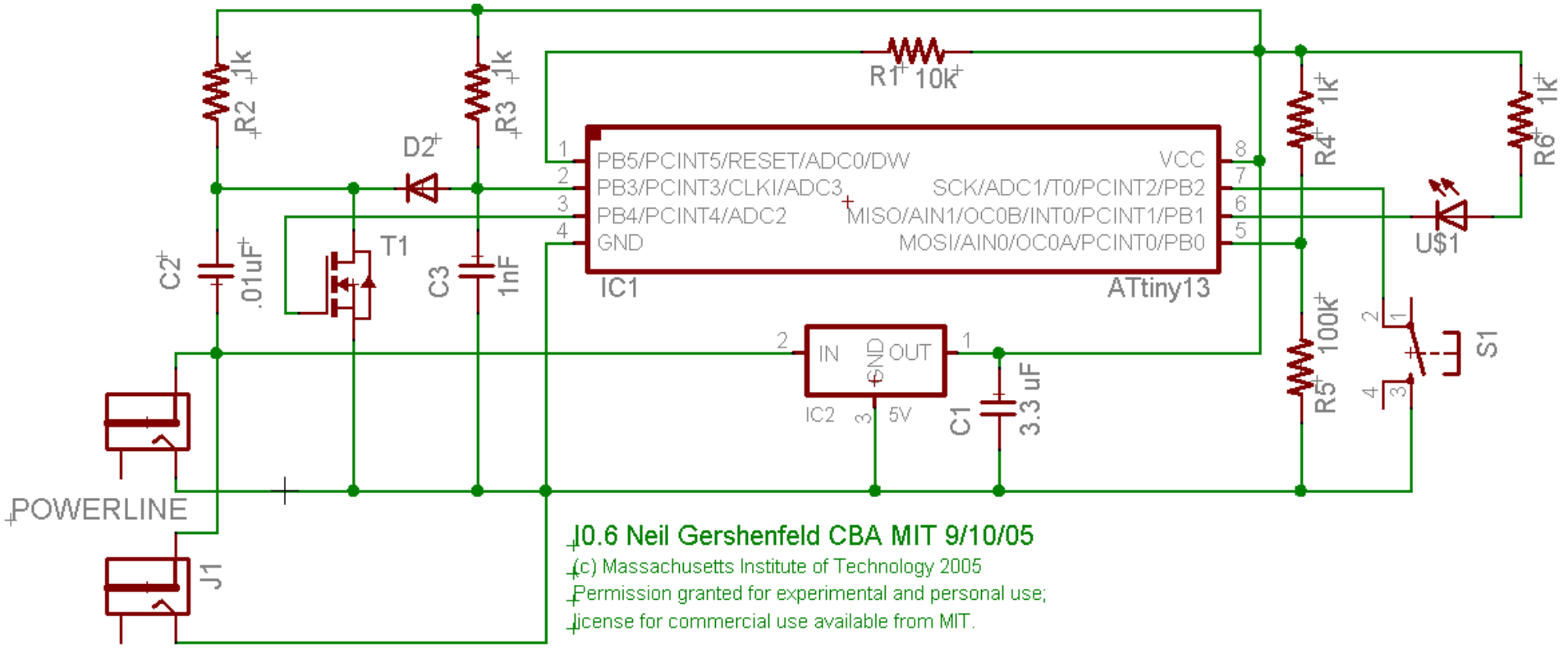


Networking

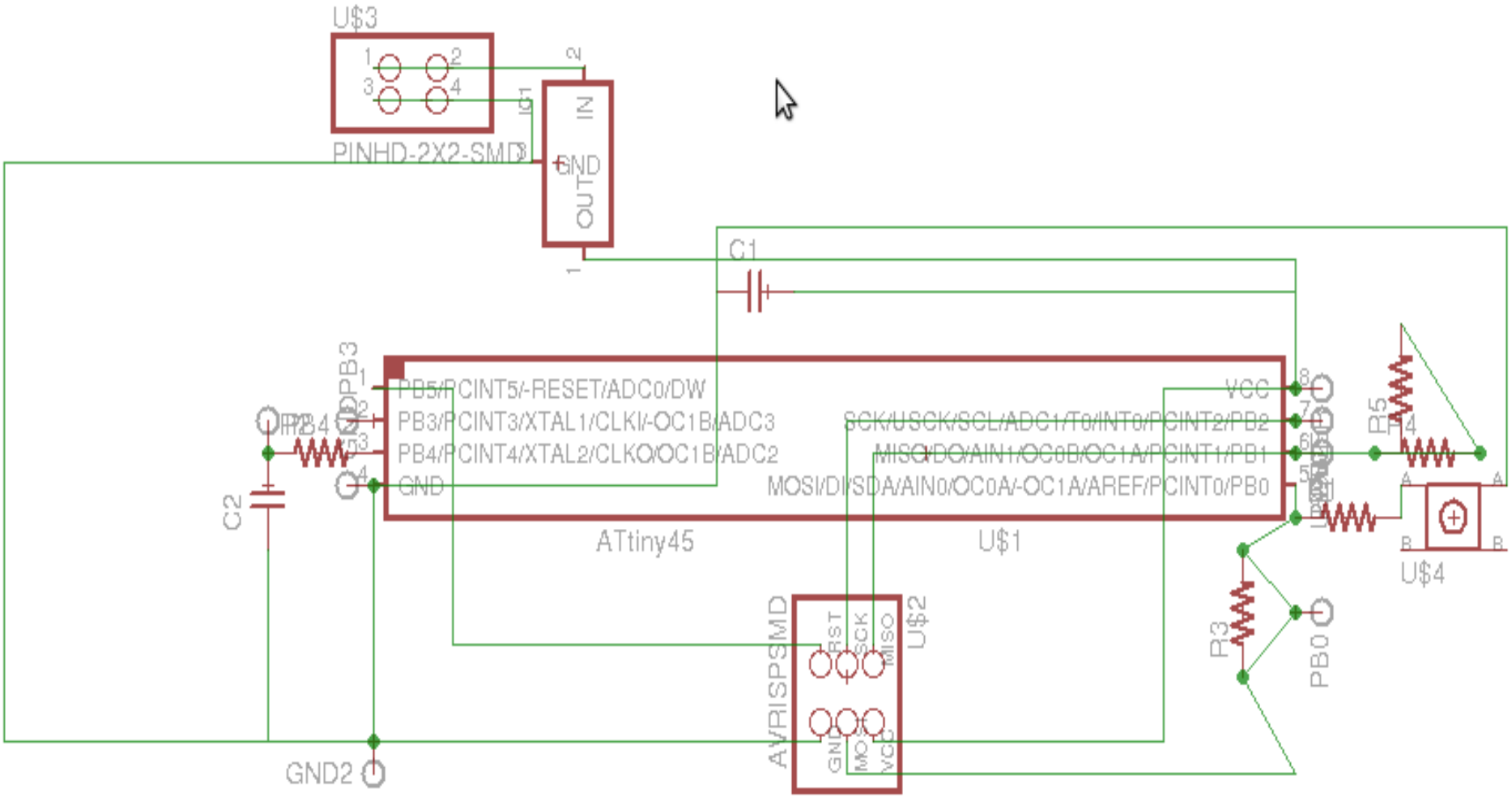
Internet 0! Blatantly ganked from Neil Gershenfeld's presentation:



The I0 board


R2 1k
R3 1k
R1 10k
R4 1k
R6 1k
D2
T1
C2 .01uF
C3 1nF
PB5/PCINT5/RESET/ADC0/DW
PB3/PCINT3/CLKI/ADC3
PB4/PCINT4/ADC2
GND
VCC
SCK/ADC1/T0/PCINT2/PB2
MISO/AIN1/OC0B/INT0/PCINT1/PB1
MOSI/AIN0/OC0A/PCINT0/PB0
IC1
ATtiny13
U$1
S1
R5 100K
IN
GND
OUT
IC2
5V
C1 3.3 uF
POWERLINE
J1
I0.6 Neil Gershenfeld CBA MIT 9/10/05
(c) Massachusetts Institute of Technology 2005
Permission granted for experimental and personal use;
license for commercial use available from MIT.

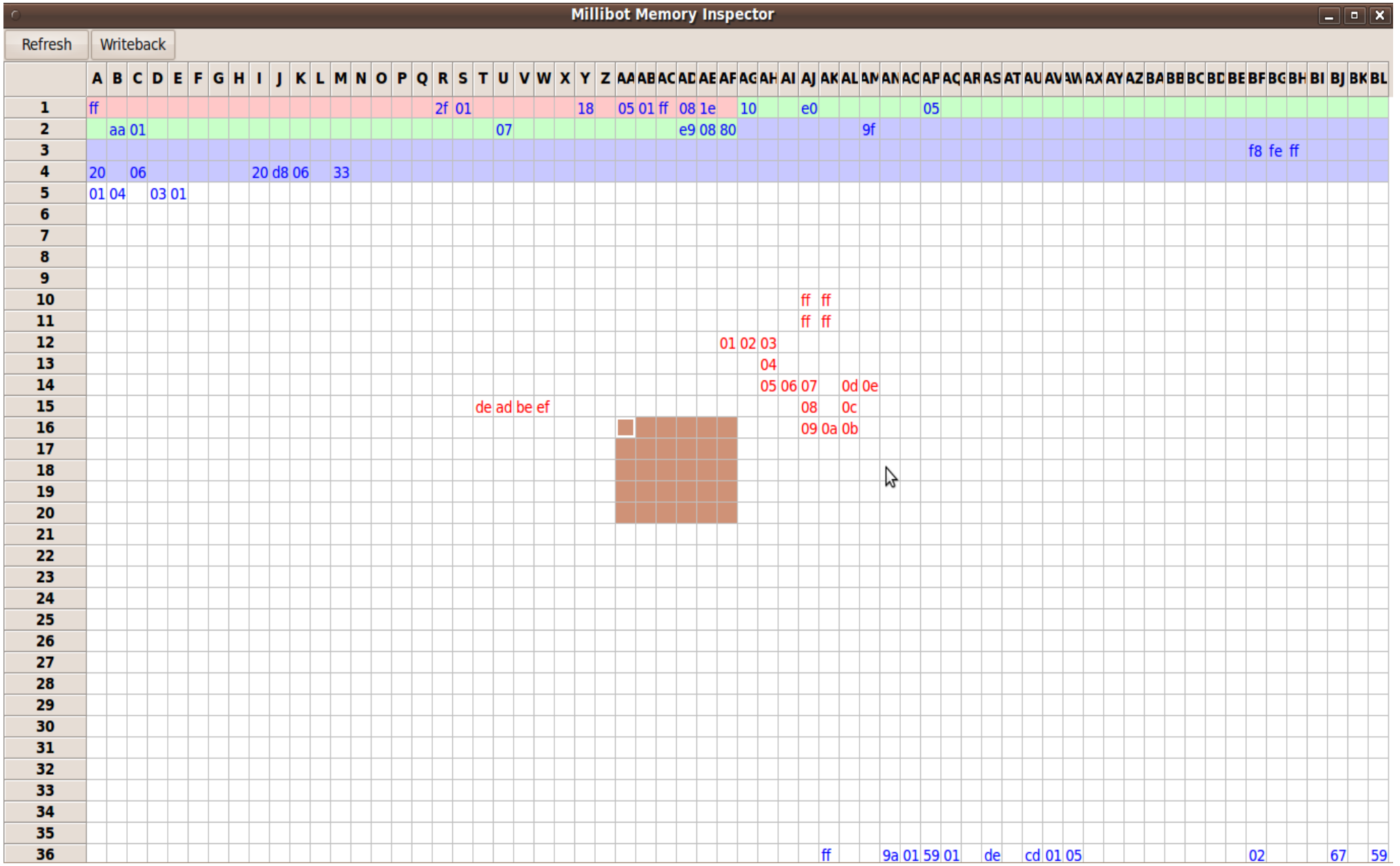
MicroSD WAV player



Controlling large robots:

Credit to: Skylar Tibbets, Ara Knaian, David Dalrymple, Forrest Green, Peter Schmidt-Nielsen, Neil Gershenfeld, at the MIT Center for Bits and Atoms.

Debugging:



Links

Toolchain:

An assembler that supports a variety of AVRs: http://www.avr-asm-tutorial.net/gavrasm/index_en.html
For those who use Windows, Atmel offers a very nice IDE (that uses avr-gcc under the hood):
http://www.atmel.com/dyn/Products/tools_card.asp?tool_id=2725

For those on a member of the UNIX family, and who wish to use C:
avr-libc: http://www.nongnu.org/avr-libc/
avrdude: http://savannah.nongnu.org/projects/avrdude

If you're on Ubuntu, this is even easier: sudo apt-get install avrdude avr-libc binutils-avr gcc-avr

Other people's code examples:

A good source of basic code examples is:
Neil Gershenfeld's page: http://fab.cba.mit.edu/about/fab

Unfortunately, Neil uses his own format to represent the boards, but fortunately gives a program to convert these ".cad" files into toolpaths of various formats (Gerber, RML, g-code, etc.).

ELM-Chan's page: (From whom we grabbed the code for the attiny85 music player): http://elm-chan.org/

Our examples:

A library for interfacing with HD44780 LCD displays: liblcd.tar
Will run on essentially any ATTiny or ATMega with enough pins.
(Written by Asa Oines and Peter Schmidt-Nielsen)

Soon to be up: Eagle files for ELM-Chan's wonderful music player, as designed by Asa Oines.